Baselining the New GSFC Information Systems Center: The Foundation for Verifiable Software Process Improvement

A. Parra, D. Schultz, J. Boger, S. Condon,
CSC
R. Webby, M. Morisio, D. Yakimovich,
J. Carver, M. Stark,
University of Maryland
V. Basili,
Fraunhofer Center Maryland and University of Maryland
S. Kraft,
NASA/GSFC

Abstract

This paper describes a study performed at the Information System Center (ISC) in NASA Goddard Space Flight Center. The ISC was set up in 1998 as a core competence center in information technology. The study aims at characterizing people, processes and products of the new center, to provide a basis for proposing improvement actions and comparing the center before and after these actions have been performed. The paper presents the ISC, goals and methods of the study, results and suggestions for improvement, through the branch-level portion of this baselining effort.

Introduction

At the beginning of 1998, a major reorganization of software engineering functions took place within the NASA Goddard Space Flight Center. A new "Information Systems Center" (ISC) was created with the objective of concentrating and consolidating Goddard's Information Technology (IT) capabilities into one organizational unit.

Within the aegis of this new organization, sits the Software Engineering Laboratory (SEL) [1,7], a twenty-three years old consortium of process and product improvement specialists from three organizations: NASA Goddard itself, the University of Maryland and Computer Sciences Corporation. The SEL had previously focused most of its efforts within the Flight Dynamics Division (FDD), performing process and product improvement studies and software engineering experiments. With the reorganization of software activities at Goddard, its scope now expands to the entire ISC. Therefore there was a need to better understand the wider context that the SEL now found itself within.

Consequently, a “baseline” study was initiated by the SEL in April 1998. The aim of the baseline was to characterize or profile the ISC in terms of its people, processes and products. Each branch and many teams within the ISC were studied for the purpose of completing an *initial* baseline study. We emphasize the word “initial” to indicate that this study is not a detailed baseline in the sense of capturing extensive focussed data about one aspect of the ISC’s operations. Rather it is a baseline that will provide an overall high-level profile of the new organization.

Many previous baselines have been conducted within the FDD, as well as at the level of Goddard Code 500 [4], Goddard as a whole [5] and NASA as a whole [6]. The questionnaires developed by the baselining team were heavily based on these earlier studies to enable comparison. Where practical, this paper will compare data from ISC with earlier studies.

This paper documents preliminary data and observations that the SEL has made in baselining the ISC. The ultimate goals of the baselining study are to identify areas for process and product improvement of benefit to Goddard, as well as interesting and novel research areas to pursue. This paper will begin by elaborating upon the goals of the study. It will continue by describing the methods adopted (and their constraints), the data collected, and the preliminary results of the work. The paper concludes with some recommendations for ISC and suggestions for future work for the SEL.

The ISC

Quoting from the ISC home page [8]:

> “The Information Systems Center (ISC) is an innovative center of expertise in the implementation of seamless, end-to-end information systems in support of NASA programs and projects, and specifically the GSFC Earth Science, Space Science and Technology focus areas. The ISC provides leadership and vision in identifying and sponsoring new and emerging information systems technologies.”

The ISC is organized in eight branches, each with a unique function. Refer to Figure 1for the organization structure of ISC and Table 1 for the associated products and services. The meaning of boxes line styles will be explained later. The work is organized in various manners: within these branches exist teams that are producing software products and services, there are personnel (and sometimes teams) matrixed to other ISC branches or other Codes at GSFC, and there are cross-branch teams that serve all the ISC with representation from the branches. The detailed organizational structure is explained in [3].

Certain terminology (noted in Italics) is used in this environment and in this paper, especially terminology related to organizational structure. Basic organizational structure is broken down from highest level to lowest, GSFC is divided into 9 *directorates*, including the Applied Engineering and Technology Directorate (AETD), within that there are 5 *Centers*, including the Information Systems Center, within that the eight *branches* mentioned above, within those branches, *teams* of individuals supporting *projects*, such as the Earth Observing System (EOS). Sometimes a person or persons is *matrixed* from one organizational entity to another, so that one group manages the work, while the person(s) maintains their original organizational alliances.

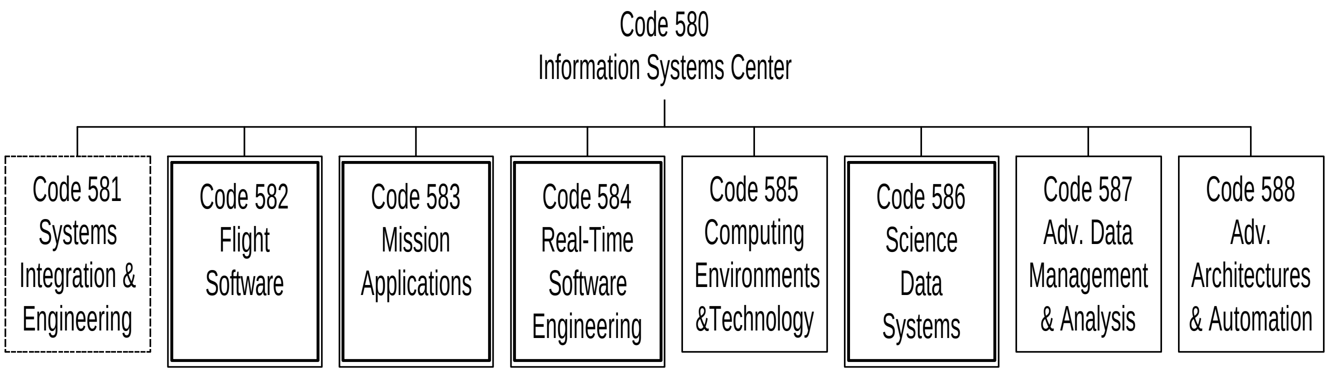

Figure 1 - Organizational Structure of the ISC

Branch Code	***Branch Name***	***Products/Services***
581	Systems Integration and Engineering	End-to-end data systems engineering of ISC mission systems development activities
582	Flight Software	Embedded software products for on-board data handling; management and control of flight hardware
583	Mission Applications	Off-line mission data systems (command management, spacecraft mission planning and scheduling, science planning, etc.)
584	Real-Time Software Engineering	Tools and services in support of information management. Real-time ground mission data systems for I&T and on-orbit ops (e.g., s/c command and control, launch, and tracking services)
585	Computing Environments and Technology	Tools and services in support of information management. Hands-on system administration, network management, WWW applications
586	Science Data Systems	Data processing, archival distribution, analysis and information management for science data systems
587	Advanced Data Management and Analysis	Advanced concept development for archival, retrieval, display, and dissemination of science data
588	Advanced Architectures and Automation	Technology R&D focused on space-ground automation systems and advanced architectures

Table 1. Products and Services of the ISC Branches

Goals for Baselining

The major objective of the baselining study is to gain an understanding of the ISC as to allow us to identify areas for process and product improvement. The philosophy behind the effort is to characterize and understand the new organization before attempting to introduce any new technology or process improvements. From the understanding, we seek to find a basis to assess improvements, which can then be packaged for wider integration into the business. Figure 2 highlights the role of baselining (the bottom rectangle) in the broader context of process and product improvement according to the Experience Factory paradigm [1].

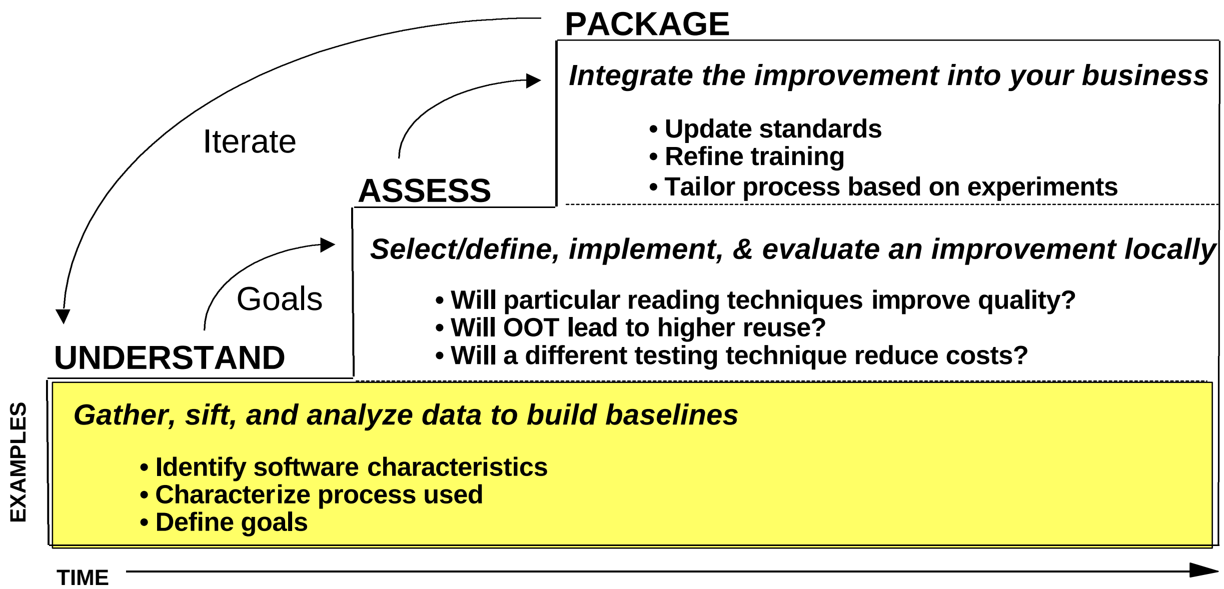

Figure 2 - Role of Baselines in Process and Product Improvement

Methods Used

The following methods, already used in the COTS Study [9], were used.

First, a number of questions and measures have been developed, starting from the high level goals and using the Goal Question Metric (GQM) approach [2], to collect information about ISC's processes, products and people. They gather both quantitative and qualitative information – some of the data are numeric and highly factual (e.g. staff numbers), whereas other data represent informed opinion (e.g., expectations of future change). The aim is to be able to characterize the software products, processes and people within the organization, with adequate qualitative context to meaningfully interpret the hard quantitative data.

Questions and measures have then been organized in a questionnaire and a structured interview [10]. The interview being constrained to no more than 30 – 45 minutes covered the qualitative data. The questionnaire was devoted to quantitative data that were less subject to interpretation.

To enforce consistency, guides for filling questionnaires and performing interviews were developed too [10].

After validating these tools with pilots, they were used to collect data from branch heads and team leaders. The process was the following.

During the interview, the Interviewer asks questions following the outline of the Interview Guide. The Scribe takes notes and employs a tape recorder, if acceptable to the Interviewee, to aid in preparation of the interview report. The Interviewee is told that the result of the interview is the interview report, which will not be considered final until the Interviewee had read and approved it. At the end of the interview the Scribe may ask some clarification questions. The Interviewer gives a copy of the Questionnaire, which asks questions of a detailed, numeric nature that don't lend themselves well to open-ended, face-to-face discussion to the Interviewee, and requests that the Questionnaire be completed within two weeks.

After the interview, the Scribe prepares an interview report, consisting of brief summaries of the Interviewee's responses to the questions on the standard Interview Guide. The Interviewer reviews the notes. Once reviewed they are sent to the Interviewee for concurrence. At this stage of the process, the interview report is considered approved. Tape recordings were not kept as the approved interview report serves as the result of the interview.

At the end of the initial interview, the Interviewer schedules a follow-up interview. The purpose of the follow-up is to go over the questionnaire that the interviewee has completed, and resolve any items where either the questions weren't clear to the interviewee, or the responses are unclear to the interviewer.

About the data

The baseline study collects data at two levels within the ISC: the branch and team levels. The current status of the study is that we have completed the branch data collection and analysis, and are currently finalizing the team-level data collection and the team-level analysis is in progress. Therefore this paper will only report on the results from the branch-level data.

The branch-level data were collected from the management of each branch. Our aim at the branch-level data collection stage was to build an overall characterization of the organization, with a wide range of factors (e.g. process, people, and product) considered. The intent is that we will perform more detailed baselines on specific factors in a subsequent study, as and when more accuracy is required.

The consequence is that the data reported in this paper have varying degrees of reliability. In some cases, they are actual data (e.g. head count). In other cases, they may be derived data. For example, a question asking how much effort was spent on software maintenance versus development was sometimes answered by managers going through their roster and counting how many people did maintenance versus development. In other cases, the data may represent only “guesstimates”. Sometimes we asked questions seeking data that they do not collect, so they had to estimate. In all cases, we are dealing with a new organization, so there is not a body of historical data, or even established data collection procedures in many cases.

As we analyze the data, we will report on the expected reliability.

Findings

Domains

Figure 3 presents a depiction of sample application domains in the ISC, in contrast to the more focused domains of the FDD. Whereas the FDD was primarily concerned with attitude, orbit and mission planning applications, the ISC must now be concerned with such diverse pursuits as science data visualization and embedded flight software. The new ISC is a much more heterogeneous organization than the FDD, so the need to understand the context of the data collected is paramount. Direct comparison of branch to branch will be meaningless without an appreciation of the context within which the data were collected.

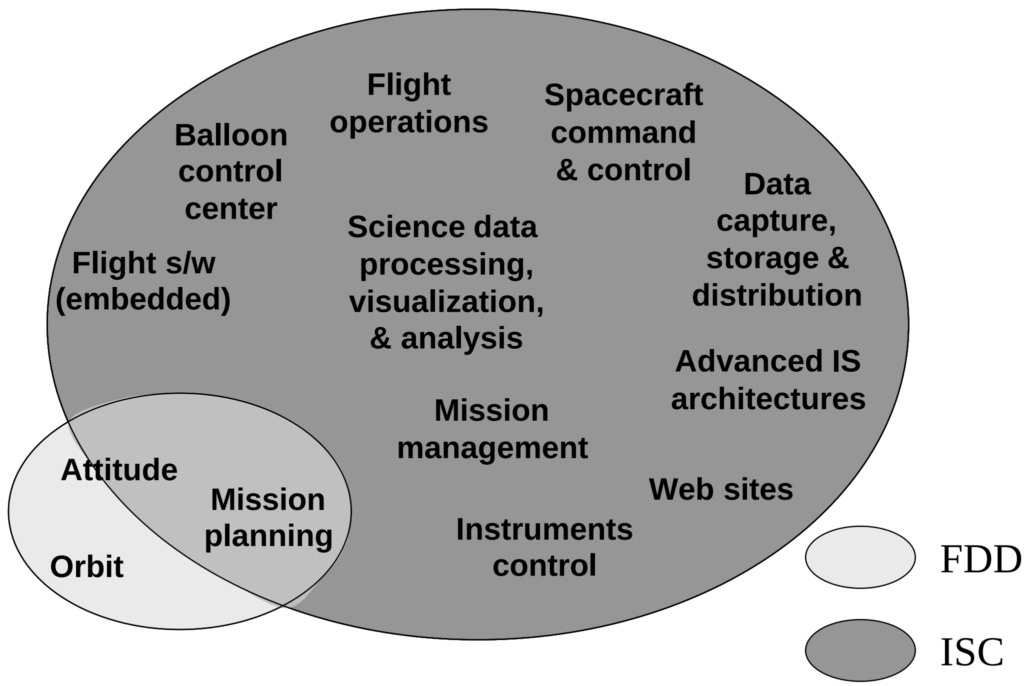

Figure 3. Sample Application Domains in ISC and FDD

Domains and organization

As mentioned above, the Information Systems Center is organizated into eight branches. Figure 1 shows the basic organizational structure of the ISC. We have found that several branches appear to have a functional domain focus (e.g. flight software), specifically these are 582, 583, 584 and 586, designated in Figure 1 with double borders. Those are contrasted with branches that deal primarily with technology domains (e.g. advanced architectures), specifically 585,587 and 588. Code 581 is probably neither in the technology nor functional camp, they deal primarily with the management of systems integration activities, this uniqueness is indicated in Figure 1 with a dashed border.

Matrixing and projects common to branches

In the questionnaire, branch management were asked to list the projects with which their branch was involved. Figure 4 presents the common projects by branch. These are larger projects such as the Hubble Space Telescope (HST) or Landsat-7, where several branches are involved. Another question was the number of staff belonging to the branch but working outside it (or matrixed). On average, 63% of ISC staff is matrixed. Both facts above suggest that the organisation by branches is in some sense virtual, while the projects rather than the branches control the process. This was also confirmed by comments from branch managers. An implication of this for the SEL is that to introduce any process improvement, it would appear necessary to consider how to influence the project to adopt the new technology.

	581	**582**	**583**	**584G**	**584W**	**585**	**586**	**587**	**588**
HST	●	●		●			●		
SMEX	●	●							
EOS	●	●	●				●	●	
EO-1		●	●			●			
ISTP	●			●					
Imp-8	●							●	
Landsat7			●			●	●		●
JSWITCH			●	●					
ULDB					●		●		

Figure 4 - Common Projects by Branch

Characterization of branches

Figure 5 presents the variation in staff numbers by branch. The total number of civil servants in ISC is 249, based on an aggregation of the questionnaire data. This total has been verified by a check against the overall ISC roster. The total number of contractors in ISC is over 308 – the exact number is difficult to determine because some branches were unable to specify their exact number of contractors[1].

[1]Staffing Numbers - The count of civil servants and subcontractors working for a branch or team is not unique, as they can report to an entity (say the team) but be paid by another (another team or branch or project). Most interviewees did not have both data, and reported the best estimate they had. An effort to collect the most accurate data is underway and will be reported in the ISC Baselining final report.

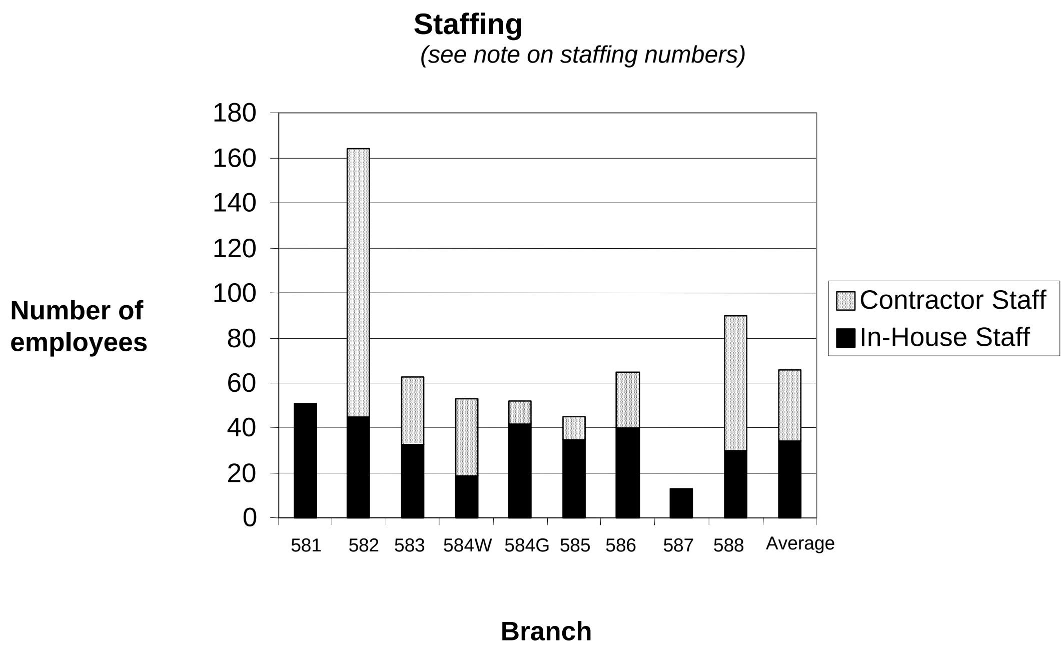

Figure 5 - Staff Numbers by Branch

Most notable here is that there is one very large branch (582), more than 2/3's of its personnel are contractors; one very small branch (587), with no contractors whatsoever; and the rest are mid-sized.

It is worthwhile to compare these figures to the SEL's 1992 baseline of Code 500 [4]. Code 500 at that time contained responsibility for most of the same functional and technology domains that the ISC contains today. Code 500, however, did not employ all of the GSFC software personnel working in these functional and technology domains; the Engineering Directorate (Code 700) employed some of them. On the other side of the balance sheet, however, we must note that some of the 1992 employees of Code 500 were analysts and other "non-software" types. These personnel were largely transferred to "Centers" other than the ISC in the recent GSFC reorganization. With these differences between the Code 500 of 1992 and the ISC of today kept in mind, let us proceed. In the 1992 baseline of code 500, it was found that approximately 1,600 of 5,000 staff (including contractors) were performing software-related functions (development, maintenance, etc). The FDD had 700 staff, of which 250 were in software. This comparison (see Figure 6) indicates that the ISC has approximately twice as many IT-related staff as FDD. However, they are significantly smaller in size than were the code 500 software people in 1992.

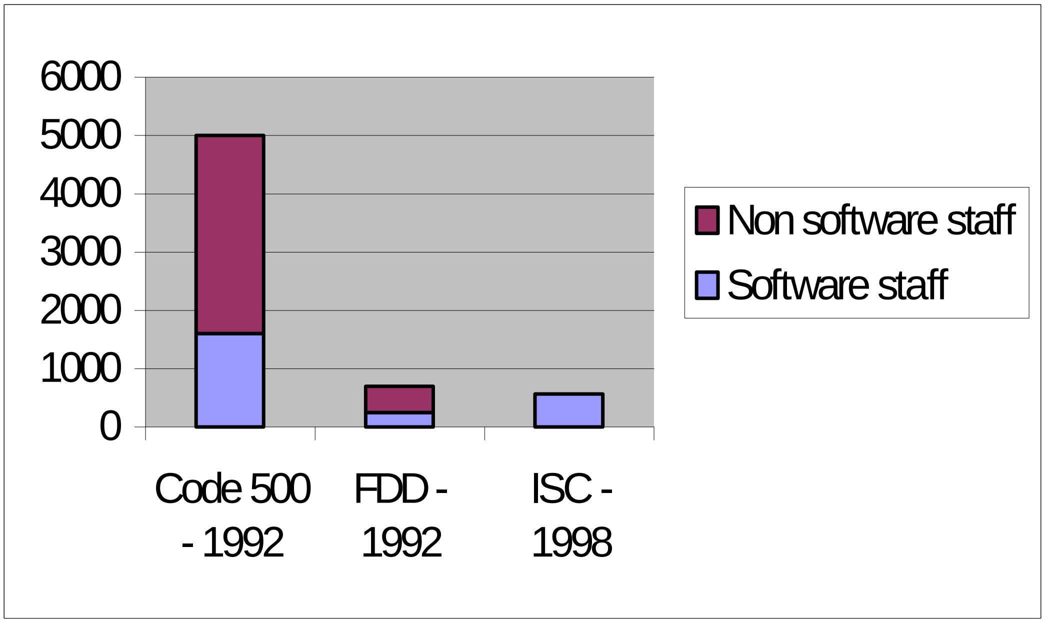

Figure 6 – Code 500, FDD and ISC staff

Branch management was also asked to estimate effort distribution within three categories: Development, Maintenance and Other. The results for this question are shown in Figure 7. The average is weighted for head-counts in the respective branches. Notable contrasts here are 581's large amount of "other" activity – as a systems integration management branch they do hardly any software development themselves. Also notable is 584 (Goddard real-time software)'s large maintenance effort relative to development effort, and 586 (science data systems)'s large development effort relative to maintenance.

In comparison with the code 500 baseline, maintenance effort in the code 500 was a lower proportion of total effort (24%) as opposed to ISC's 35% of effort devoted to maintenance. This is probably explained by the smaller amount of legacy code that the ISC is responsible for maintaining, in comparison to code 500.

Figure 8 turns our focus on software development effort alone, broken into the activities 'requirements analysis', 'design', 'coding', 'testing' and 'other'. It is apparent that at this macro process level, there is relatively little difference between ISC's average development effort distribution and that of the 1992 FDD. The ISC do a little more requirements, but that is the only major difference. Again, we should stress that these data are management estimates, not the actual recorded effort for each employee. In some cases, managers used heuristics such as counting the number of testers in the organization to come up with the proportion of testing being done. But did this then account for developers' unit testing? We do not know.

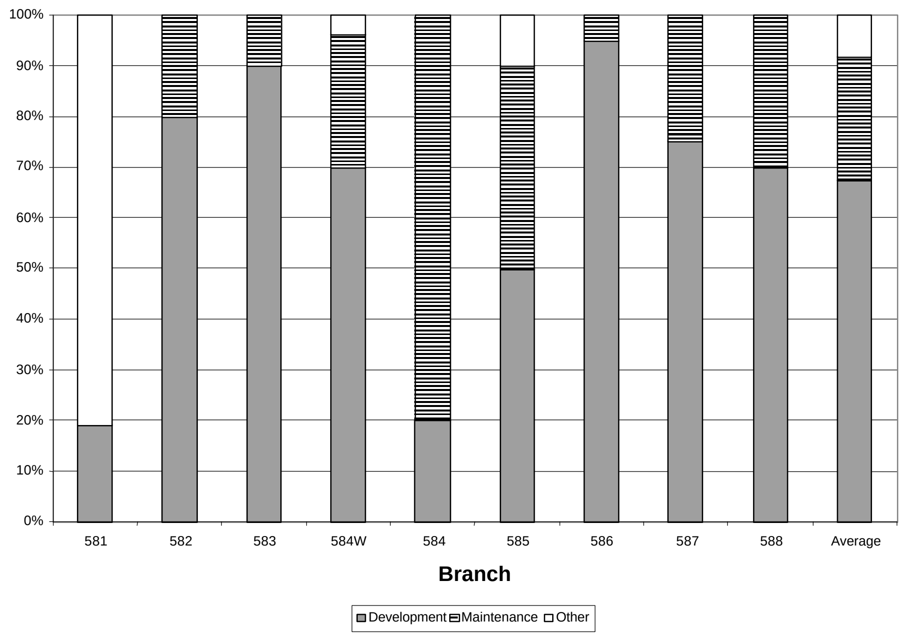

Figure 7 - Overall Effort by Branch

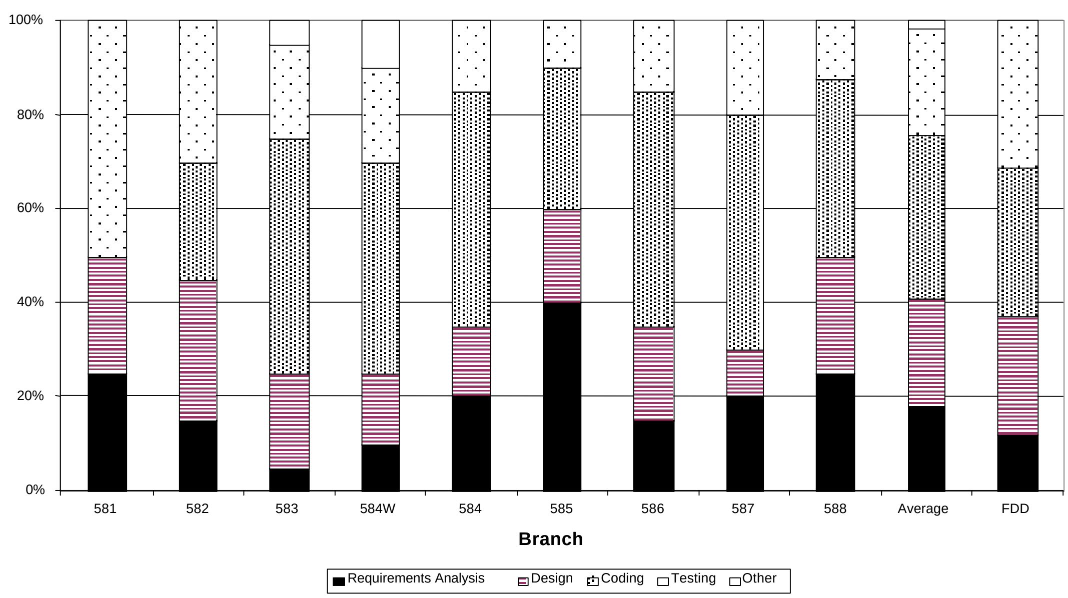

Figure 8 - Development Effort by Branch

One possible interpretation of this data is that organizations that are more outwardly focused, have had to put more effort into the requirements stage (and hence proportionally less in other areas). Code 585 (science data systems) is an example of this – much of their work is for the science community as a whole, a fairly diverse and remotely located user population. Code 583 (mission applications) has a much more defined user base and develops software such as off-line mission scheduling systems that can be precisely specified more easily up-front.

Some further observations about process, product and knowledge levels. Note that all branch averages are weighted by the number of staff in the branch.

- The percent of branches (including contractors) using "defined, written, advocated software processes" varied from 10-95%, with an average of 45%
- The percent of branches (including contractors) using "software standards" ranged from 0-95%, with an average of 57%
- The number of COTS products used varied from 2-10 with an average of 5.1. Note that these figures are probably deflated due to some branches listing "DBMSs", or "lots" in response to this question.
- Overall the use of C++, Java and Ada for new development is increasing, relative to Assembly, Fortran and C. 12 languages are used across ISC as a whole.
- The most significant causes of errors in operational software were (in the following order of importance): 'changing requirements', 'missing requirements', 'misinterpreted requirements', 'coding errors', 'interfaces', 'design errors' and 'environment problems'.
- Most branches consider themselves well-informed about 'prototyping', 'object-oriented technology', 'inspections/walkthroughs', and 'COTS Integration'
- Most branches consider themselves to have relatively little knowledge about 'formal methods' and 'defect causal analysis', except 586 science data systems
- Most branches consider themselves to have relatively little knowledge about 'information hiding' except 584W real-time systems (Wallops)
- All branches consider themselves to have relatively little knowledge about 'Cleanroom techniques'.
- Only three branches produce 'lessons learned' documents at the end of a project. Interestingly, one of these (584W) also produce a document called 'a day in the life' which serves to portray a typical day's activities for a developer. This is considered useful for training purposes.

In the process improvement area, several of the branches have ongoing activities:

- Code 581 is funding this ISC baselining study, and is also leading the ISO 9000 ISC certification. It is also pursuing an effort to define a core metrics set with the SEL and Code 300.
- Code 582 is encouraging reuse of both flight software and ground simulators, is looking into additional opportunities for automatic code generation, and is pursuing the use of COTS.
- Code 583 has implemented the CORE TEAM approach, which is a type of process improvement, and some parts of the branch are involved in some level of data collection.

- Codes 584 and 587 are currently defining their processes, as a prelude to improving them. Code 584 expressed a desire to define a multi-level process structure, to facilitate modularization of processes.
- Code 585, although it has not initiated a formal process improvement program, is using guidelines in certain areas. The Code 585 personnel prefer to use guidelines, rather than standards, because of the greater flexibility that guidelines provide.
- Code 586 is engaged in process management activities, including implementation of ISO 9001.
- Code 588, for the most part, has not initiated any process improvement activities; they are, however, currently working on a Technology Management Plan that is oriented toward ISO 9000. Code 588 is also trying to move the designation of their ultimate customer organization earlier in the process of making a system operational.

Analysis and further activities

The ISC is a new organization that supports many of the key projects at NASA Goddard. It is divided into management, technology and functional branches that represent a wide variety of technical and functional domains. Here we try to summarize the main results of the baselining effort and their implications for further SEL activities.

Diversity

The preliminary results of this baseline show that each branch is very different in terms of personnel, process and product characteristics. The variations in effort distribution, languages used, and products developed by the different branches provide surface indications of the diversity among the branches. The implications are that it will not be possible to apply the same models for cost and quality to each branch, as we could do to some extent within the more homogeneous FDD. To understand how cost and quality relate, we must study them in the context of each branch, team and/or project. Then, each model must be constructed and calibrated to the given context in question. The development of different models however is not the only challenge; these models must be capable of integration so that aggregated information can be meaningfully provided for the whole of ISC.

The NASA Core Software Metrics Initiative

The SEL and GSFC/NASA's Software Assurance Technology Center (SATC) [11] are currently pursuing an initiative to define and implement a core set of software metrics, common to the whole of NASA. For well over a year these two GSFC organizations have been working together to define a core set of metrics.

The baselining has confirmed that there is an essential need for core metrics within the ISC. Due to the diversity of the ISC, branches, teams and projects use different reporting units for metrics such as product size, effort and defects. The core metrics initiative defines a set of metrics capable of being used in different contexts, yet capable of providing a common abstraction level to allow aggregation at the ISC level. This is essential not only for monitoring purposes, but also for the model building needs mentioned above.

At this time, a draft version of the Core Metrics set, developed by the SEL and SATC, is currently under review by the NASA Software Working Group. At the time this paper is written the SATC and SEL web pages do not specifically call out the Core Metrics, in future that information should be assessable through SATC and SEL web pages [11,12]. An experiment within the ISC to validate these Core Metrics would serve both the NASA Core Metric Initiative and the ISC's proactive drive toward process and product awareness and improvement.

Matrixing

The ISC is organized in branches and teams, but branch and team staff work, at 63% on average, on projects outside the scope of ISC, managed and funded by NASA Codes other than 500. In particular, 95% of the staff belonging to Code 582 is matrixed outside ISC. This is not surprising, as the ISC is meant to offer IT services to all of GSFC and NASA. However, a number of issues are raised.

- System and software engineering. Many projects where matrixed staff works are system projects where software is only a part. The system issues (processes, technologies, interfaces) should be taken into account in software processes too.
- Ownership of processes and rights to modify. When projects are funded and ruled outside ISC, ISC may or may not be free to decide on processes, standards, and organizations to be used.
- Diffusion of information. Matrixed personnel could physically work outside ISC, with increased difficulties in communication and diffusion of information about the SEL and technology transfer or software process improvement projects.

The SEL could try to understand in more depth these issues with further studies. However, it seems that, for the purposes of assessment, characterization, and model building, the team and the projects are the more suitable units to be considered. This implies that, as projects and teams are volatile, with a life span of months, measures and models should be highly versatile and adaptive.

Also, the concept of Experience Factory, defined and used by the SEL in the past years, could need some adaptation. Several levels of experience, and several levels of learning loops, can be identified: at the individual, team, branch and ISC levels.

Finally, if projects and teams are volatile, and branches are virtual, individual persons are the most stable and valuable resources to base process and product improvement on. Approaches such as Watt Humphrey's Personal Software Process (PSP) could be used and adapted to the ISC context. Specifically, the PSP does not consider sharing experiences and improvements with peers, and should be extended in this direction to integrate concepts from the Experience Factory.

COTS

All branches report the use of COTS. The SEL should support teams and branches in COTS related activities: evaluation and selection, testing and certification, interaction with producer, documentation and diffusion of information. The SEL's experience in COTS processes will be of benefit to the ISC and the diversity of the ISC offers opportunities for case studies to further validate the COTS process model [9]. This study concluded with recommendations for further work to build cost models, risk analysis, and process models. Since, COTS remains a buzzword with different meanings for different people. Another action for the SEL is the definition of a set of terms and classification tools for the different concepts and artifacts currently considered under the umbrella term COTS.

Finally, COTS should be considered in the broader context of reuse and related technological and organizational issues: domain analysis and engineering, product line engineering, reusable libraries, frameworks, design patterns, mechanisms and standards (Com, Corba, Active-X, Java RMI, Java beans, etc.).

Internal technology transfer

There would seem to be opportunities for greater synergies within ISC to do internal technology transfer so that the advanced technologies and research efforts of branches 585, 587 and 588 are successfully transitioned into practice in branches 582, 583, 584 and 586.

The past work of the SEL within Goddard has shown the need to understand, assess and package technology to insure its successful introduction. Possibly the SEL in code 581 can play a role in furthering a controlled and systematic transfer of this technology to the functional branches, as well as helping insure that the advanced technology branches work in relevant areas amenable to future technology transfer.

The SEL could assist by defining a methodology to evaluate if and how a technology successfully applied in one context (branch, team, project) can be transferred to another context.

Reuse and frameworks

Several products in ISC are developed and mantained for years and possibly customised in different versions. The overall cost of a product during the complete service cycle can be decreased by technologies such as architecture and framework-based reuse. For example Code 582 (flight software) is exploring this road by developing a new architectural design for on-board shuttle navigation control.

The SEL could offer support to organize, measure and document such efforts with two main goals. Promote the success of the reuse effort inside a branch. And acquire methodological experience to replicate the same effort in other branches (see also the Internal Technology Transfer subsection).

Requirements instability

Requirements, and specifically requirements instability, are a common source of problems for ISC teams. Several lines of intervention are available for the SEL:

- Experimentation with novel techniques for requirements capture and management.
- Adaptation of and experimentation with of techniques for early detection of defects in requirements, such as requirement reading techniques.
- Adaptation of and experimentation with new lifecycles for early verification of requirements, such as prototyping, iterative lifecycles, joint application development.

Acknowledgements

This work was funded by NASA grant NCC-5170, and the following NASA Contracts: CNMOS and CSOC.

References

[1] V. R. Basili, G. Caldiera, F. McGarry, R. Pajerski, G. Page, S. Waligora, The Software Engineering Laboratory - an Operational Software Experience Factory, *International Conference on Software Engineering*, May, 1992, pp. 370-381.

[2] R. Basili, H. D. Rombach, The TAME Project: Towards Improvement-Oriented Software Environments, *IEEE Transactions on Software Engineering*, vol. SE-14, no.6, June 1988.

[3] Kea H., Goddard's New Integrated Approach to Information Technology, 23rd Software Engineering Workshop, Nasa/GFSC, December 1998.

[4] NASA, Profile of Software Within Code 500 at Goddard Space Flight Center, Technical report R01-92, 1992.

[5] NASA, Profile of Software at the Goddard Space Flight Center, Technical report RPT-002-94, June 1994.

[6] NASA, Profile of Software at NASA, Technical report RPT-93, December 1993.

[7] NASA, An Overview of the Software Engineering Lab, Technical report SEL-94-005, December 1994.

[8] NASA/ISC, The ISC home page, http://isc.gsfc.nasa.gov/default.htm.

[9]NASA/SEL, SEL COTS Study, Phase 1, Initial Characterization Study report, SEL-98-001, August 1998.

[10] NASA/SEL, ISC Baselining documentation, http://sel.gsfc.nasa.gov/doc-st/tech-st/sew23/baselining.htm

[11] NASA/SATC, The SATC home page, http://satc.gsfc.nasa.gov/

[12] NASA/SEL, The SEL home page, http://sel.gsfc.nasa.gov/